

Manufactured Solution for 3D Euler equation using Maple*

Kemelli C. Estacio-Hiroms

April 13, 2010

Abstract

This document presents the source terms for code verification for the numerical solutions of the 3D Euler equations. Such source terms are obtained by the Method of Manufactured Solutions (MMS) using the analytical solution for density, velocity and pressure, proposed by (Roy et al., 2002).

1 3D Euler Equations

The 3D Euler equations in conservation form are:

$$\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad (1)$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(\rho uw)}{\partial z} = 0 \quad (2)$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu)}{\partial x} + \frac{\partial(\rho v^2 + p)}{\partial y} + \frac{\partial(\rho vw)}{\partial z} = 0 \quad (3)$$

$$\frac{\partial(\rho w)}{\partial t} + \frac{\partial(\rho wu)}{\partial x} + \frac{\partial(\rho wv)}{\partial y} + \frac{\partial(\rho w^2 + p)}{\partial z} = 0 \quad (4)$$

$$\frac{\partial(\rho e_t)}{\partial t} + \frac{\partial(\rho u e_t + pu)}{\partial x} + \frac{\partial(\rho v e_t + pv)}{\partial y} + \frac{\partial(\rho w e_t + pw)}{\partial z} = 0 \quad (5)$$

where the Equation (1) is the unsteady term (mass conservation), Equations (2)–(4) are the nonlinear convection term in the x , y and z direction (momentum), and Equation (5) is the energy. For a calorically perfect gas, the Euler equations are closed with two auxiliary relations for energy:

$$e = \frac{1}{\gamma - 1} RT, \quad (6)$$

*Work based on Roy, Smith, and Ober (2002).

$$e_t = e + \frac{u^2 + v^2 + w^2}{2}, \quad (7)$$

and with the ideal gas equation of state:

$$p = \rho RT. \quad (8)$$

2 Manufactured Solution

Roy et al. (2002) propose the general form of the primitive solution variables to be a function of sines and cosines:

$$\phi(x, y) = \phi_0 + \phi_x f_s\left(\frac{a_{\phi x} \pi x}{L}\right) + \phi_y f_s\left(\frac{a_{\phi y} \pi y}{L}\right) + \phi_z f_s\left(\frac{a_{\phi z} \pi z}{L}\right), \quad (9)$$

where $\phi = \rho, u, v, w$ or p , and $f_s(\cdot)$ functions denote either sine or cosine function. Note that in this case, ϕ_x , ϕ_y and ϕ_z are constants and the subscripts do not denote differentiation.

Therefore, the manufactured analytical solution for for each one of the variables in Euler equations are:

$$\begin{aligned} \rho(x, y, z) &= \rho_0 + \rho_x \sin\left(\frac{a_{\rho x} \pi x}{L}\right) + \rho_y \cos\left(\frac{a_{\rho y} \pi y}{L}\right) + \rho_z \sin\left(\frac{a_{\rho z} \pi z}{L}\right), \\ u(x, y, z) &= u_0 + u_x \sin\left(\frac{a_{ux} \pi x}{L}\right) + u_y \cos\left(\frac{a_{uy} \pi y}{L}\right) + u_z \cos\left(\frac{a_{uz} \pi z}{L}\right), \\ v(x, y, z) &= v_0 + v_x \cos\left(\frac{a_{vx} \pi x}{L}\right) + v_y \sin\left(\frac{a_{vy} \pi y}{L}\right) + v_z \sin\left(\frac{a_{vz} \pi z}{L}\right), \\ w(x, y, z) &= w_0 + w_x \sin\left(\frac{a_{wx} \pi x}{L}\right) + w_y \sin\left(\frac{a_{wy} \pi y}{L}\right) + w_z \cos\left(\frac{a_{wz} \pi z}{L}\right), \\ p(x, y, z) &= p_0 + p_x \cos\left(\frac{a_{px} \pi x}{L}\right) + p_y \sin\left(\frac{a_{py} \pi y}{L}\right) + p_z \cos\left(\frac{a_{pz} \pi z}{L}\right). \end{aligned} \quad (10)$$

Constants ϕ_0 , ϕ_x , ϕ_y and ϕ_z with $\phi = u, v, w, p, \rho$ for the manufactured solutions (10) for the 3D supersonic and subsonic cases are presented in Roy et al. (2002).

3 Source terms for 3D Euler equations

The source terms Q_ρ , Q_u , Q_v , Q_w and Q_{e_t} obtained through symbolic manipulation of 3D Euler equations (1) – (5), respectively, over the analytical solutions (10) are:

[illegible]

$$\begin{aligned}
& + \frac{\pi a_{vy} v_y}{2L} \cos\left(\frac{\pi y a_{vy}}{L}\right) \left\{ \left(\left[u_x \sin\left(\frac{\pi x a_{ux}}{L}\right) + u_y \cos\left(\frac{\pi y a_{uy}}{L}\right) + u_z \cos\left(\frac{\pi z a_{uz}}{L}\right) + u_0 \right]^2 + 3 \left[v_x \cos\left(\frac{\pi x a_{vx}}{L}\right) + v_y \sin\left(\frac{\pi y a_{vy}}{L}\right) + v_z \sin\left(\frac{\pi z a_{vz}}{L}\right) + v_0 \right]^2 + \right. \right. \\
& + \left. \left[w_x \sin\left(\frac{\pi x a_{wx}}{L}\right) + w_y \sin\left(\frac{\pi y a_{wy}}{L}\right) + w_z \cos\left(\frac{\pi z a_{wz}}{L}\right) + w_0 \right]^2 \right) \left[\rho_x \sin\left(\frac{\pi x a_{\rho x}}{L}\right) + \rho_y \cos\left(\frac{\pi y a_{\rho y}}{L}\right) + \rho_z \sin\left(\frac{\pi z a_{\rho z}}{L}\right) + \rho_0 \right] + \\
& + \left. \left[p_x \cos\left(\frac{\pi x a_{px}}{L}\right) + p_y \sin\left(\frac{\pi y a_{py}}{L}\right) + p_z \cos\left(\frac{\pi z a_{pz}}{L}\right) + p_0 \right] \frac{2\gamma}{\gamma - 1} \right\} + \\
& + \frac{\pi a_{vz} v_z}{L} \cos\left(\frac{\pi z a_{vz}}{L}\right) \left[w_x \sin\left(\frac{\pi x a_{wx}}{L}\right) + w_y \sin\left(\frac{\pi y a_{wy}}{L}\right) + w_z \cos\left(\frac{\pi z a_{wz}}{L}\right) + w_0 \right] \left[v_x \cos\left(\frac{\pi x a_{vx}}{L}\right) + v_y \sin\left(\frac{\pi y a_{vy}}{L}\right) + v_z \sin\left(\frac{\pi z a_{vz}}{L}\right) + v_0 \right] \cdot \\
& \cdot \left[\rho_x \sin\left(\frac{\pi x a_{\rho x}}{L}\right) + \rho_y \cos\left(\frac{\pi y a_{\rho y}}{L}\right) + \rho_z \sin\left(\frac{\pi z a_{\rho z}}{L}\right) + \rho_0 \right] + \\
& + \frac{\pi a_{wx} w_x}{L} \cos\left(\frac{\pi x a_{wx}}{L}\right) \left[w_x \sin\left(\frac{\pi x a_{wx}}{L}\right) + w_y \sin\left(\frac{\pi y a_{wy}}{L}\right) + w_z \cos\left(\frac{\pi z a_{wz}}{L}\right) + w_0 \right] \left[u_x \sin\left(\frac{\pi x a_{ux}}{L}\right) + u_y \cos\left(\frac{\pi y a_{uy}}{L}\right) + u_z \cos\left(\frac{\pi z a_{uz}}{L}\right) + u_0 \right] \cdot \\
& \cdot \left[\rho_x \sin\left(\frac{\pi x a_{\rho x}}{L}\right) + \rho_y \cos\left(\frac{\pi y a_{\rho y}}{L}\right) + \rho_z \sin\left(\frac{\pi z a_{\rho z}}{L}\right) + \rho_0 \right] + \\
& + \frac{\pi a_{wy} w_y}{L} \cos\left(\frac{\pi y a_{wy}}{L}\right) \left[w_x \sin\left(\frac{\pi x a_{wx}}{L}\right) + w_y \sin\left(\frac{\pi y a_{wy}}{L}\right) + w_z \cos\left(\frac{\pi z a_{wz}}{L}\right) + w_0 \right] \left[v_x \cos\left(\frac{\pi x a_{vx}}{L}\right) + v_y \sin\left(\frac{\pi y a_{vy}}{L}\right) + v_z \sin\left(\frac{\pi z a_{vz}}{L}\right) + v_0 \right] \cdot \\
& \cdot \left[\rho_x \sin\left(\frac{\pi x a_{\rho x}}{L}\right) + \rho_y \cos\left(\frac{\pi y a_{\rho y}}{L}\right) + \rho_z \sin\left(\frac{\pi z a_{\rho z}}{L}\right) + \rho_0 \right] + \\
& - \frac{\pi a_{wz} w_z}{2L} \sin\left(\frac{\pi z a_{wz}}{L}\right) \left\{ \left(\left[u_x \sin\left(\frac{\pi x a_{ux}}{L}\right) + u_y \cos\left(\frac{\pi y a_{uy}}{L}\right) + u_z \cos\left(\frac{\pi z a_{uz}}{L}\right) + u_0 \right]^2 + \left[v_x \cos\left(\frac{\pi x a_{vx}}{L}\right) + v_y \sin\left(\frac{\pi y a_{vy}}{L}\right) + v_z \sin\left(\frac{\pi z a_{vz}}{L}\right) + v_0 \right]^2 + \right. \right. \\
& + 3 \left. \left[w_x \sin\left(\frac{\pi x a_{wx}}{L}\right) + w_y \sin\left(\frac{\pi y a_{wy}}{L}\right) + w_z \cos\left(\frac{\pi z a_{wz}}{L}\right) + w_0 \right]^2 \right) \left[\rho_x \sin\left(\frac{\pi x a_{\rho x}}{L}\right) + \rho_y \cos\left(\frac{\pi y a_{\rho y}}{L}\right) + \rho_z \sin\left(\frac{\pi z a_{\rho z}}{L}\right) + \rho_0 \right] + \\
& + \left. \left[p_x \cos\left(\frac{\pi x a_{px}}{L}\right) + p_y \sin\left(\frac{\pi y a_{py}}{L}\right) + p_z \cos\left(\frac{\pi z a_{pz}}{L}\right) + p_0 \right] \frac{2\gamma}{\gamma - 1} \right\}
\end{aligned} \tag{16}$$

4 C codes

Files containing C codes for the source terms have also been generated. They are: `Euler_3d_e.code.C`, `Euler_3d_rho.code.C`, `Euler_3d_u.code.C`, `Euler_3d_v.code.C` and `Euler_3d_w.code.C`. The gradients of the analytical solutions have also been computed and their respective C codes are presented in `Euler_manuf_solutions_grad_and.code.3d.C`

An example of the automatically generated C file from the source term for mass conservation equation is:

```

#include <math.h>

double SourceQ_u (double x, double y, double z, double u_0, double u_x, double u_y, double u_z, double v_0, double v_x,
double v_y, double v_z, double rho_0, double rho_x, double rho_y, double rho_z, double p_0, double p_x,
double p_y, double p_z, double a_px, double a_py, double a_pz, double a_rhox, double a_rhoy,
double a_rhoz, double a_ux, double a_uy, double a_uz, double a_vx, double a_vy, double a_vz, double a_wx,

```

```

double a_wy, double a_wz, double L)
{
double Q_u;
Q_u = -p_x * sin(a_px * PI * x / L) * a_px * PI / L +
rho_x * cos(a_rhox * PI * x / L) * pow(u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) +
u_z * cos(a_uz * PI * z / L), 0.2e1) * a_rhox * PI / L -
rho_y * sin(a_rhoy * PI * y / L) * (v_0 + v_x * cos(a_vx * PI * x / L) + v_y * sin(a_vy * PI * y / L) +
v_z * sin(a_vz * PI * z / L)) * (u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) +
u_z * cos(a_uz * PI * z / L)) * a_rhoy * PI / L +
rho_z * cos(a_rhoz * PI * z / L) * (w_0 + w_x * sin(a_wx * PI * x / L) + w_y * sin(a_wy * PI * y / L) +
w_z * cos(a_wz * PI * z / L)) * (u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) +
u_z * cos(a_uz * PI * z / L)) * a_rhoz * PI / L +
0.2e1 * u_x * cos(a_ux * PI * x / L) * (rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) +
rho_z * sin(a_rhoz * PI * z / L)) * (u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) +
u_z * cos(a_uz * PI * z / L)) * a_ux * PI / L -
u_y * sin(a_uy * PI * y / L) * (rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) +
rho_z * sin(a_rhoz * PI * z / L)) * (v_0 + v_x * cos(a_vx * PI * x / L) + v_y * sin(a_vy * PI * y / L) +
v_z * sin(a_vz * PI * z / L)) * a_uy * PI / L -
u_z * sin(a_uz * PI * z / L) * (rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) +
rho_z * sin(a_rhoz * PI * z / L)) * (w_0 + w_x * sin(a_wx * PI * x / L) + w_y * sin(a_wy * PI * y / L) +
w_z * cos(a_wz * PI * z / L)) * a_uz * PI / L +
v_y * cos(a_vy * PI * y / L) * (rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) +
rho_z * sin(a_rhoz * PI * z / L)) *
(u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) + u_z * cos(a_uz * PI * z / L)) * a_vy * PI / L -
w_z * sin(a_wz * PI * z / L) * (rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) +
rho_z * sin(a_rhoz * PI * z / L)) * (u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) +
u_z * cos(a_uz * PI * z / L)) * a_wz * PI / L;
return(Q_u);
}

```

Finally the gradients of the analytical solutions (10) have also been computed and their respective C codes are presented in Euler_manuf_solutions_grad.and.code_3d.C. Therefore,

$$\nabla \rho = \begin{bmatrix} \frac{a_{\rho x} \pi \rho_x}{L} \cos\left(\frac{a_{\rho x} \pi x}{L}\right) \\ -\frac{a_{\rho y} \pi \rho_y}{L} \sin\left(\frac{a_{\rho y} \pi y}{L}\right) \\ \frac{a_{\rho z} \pi \rho_z}{L} \cos\left(\frac{a_{\rho z} \pi z}{L}\right) \end{bmatrix}, \quad \nabla p = \begin{bmatrix} -\frac{a_{px} \pi p_x}{L} \sin\left(\frac{a_{px} \pi x}{L}\right) \\ \frac{a_{py} \pi p_y}{L} \cos\left(\frac{a_{py} \pi y}{L}\right) \\ -\frac{a_{pz} \pi p_z}{L} \sin\left(\frac{a_{pz} \pi z}{L}\right) \end{bmatrix}, \quad \nabla u = \begin{bmatrix} \frac{a_{ux} \pi u_x}{L} \cos\left(\frac{a_{ux} \pi x}{L}\right) \\ -\frac{a_{uy} \pi u_y}{L} \sin\left(\frac{a_{uy} \pi y}{L}\right) \\ -\frac{a_{uz} \pi u_z}{L} \sin\left(\frac{a_{uz} \pi z}{L}\right) \end{bmatrix}, \quad (17)$$

$$\nabla v = \begin{bmatrix} -\frac{a_{vx}\pi v_x}{L} \sin\left(\frac{a_{vx}\pi x}{L}\right) \\ \frac{a_{vy}\pi v_y}{L} \cos\left(\frac{a_{vy}\pi y}{L}\right) \\ \frac{a_{vz}\pi v_z}{L} \cos\left(\frac{a_{vz}\pi z}{L}\right) \end{bmatrix} \quad \text{and} \quad \nabla w = \begin{bmatrix} \frac{a_{wx}\pi w_x}{L} \cos\left(\frac{a_{wx}\pi x}{L}\right) \\ \frac{a_{wy}\pi w_y}{L} \cos\left(\frac{a_{wy}\pi y}{L}\right) \\ -\frac{a_{wz}\pi w_z}{L} \sin\left(\frac{a_{wz}\pi z}{L}\right) \end{bmatrix} \quad (18)$$

are written in C language as:

```
grad_rho_an[0] = rho_x * cos(a_rhox * pi * x / L) * a_rhox * pi / L;
grad_rho_an[1] = -rho_y * sin(a_rhoy * pi * y / L) * a_rhoy * pi / L;
grad_rho_an[2] = rho_z * cos(a_rhoz * pi * z / L) * a_rhoz * pi / L;
grad_p_an[0] = -p_x * sin(a_px * pi * x / L) * a_px * pi / L;
grad_p_an[1] = p_y * cos(a_py * pi * y / L) * a_py * pi / L;
grad_p_an[2] = -p_z * sin(a_pz * pi * z / L) * a_pz * pi / L;
grad_u_an[0] = u_x * cos(a_ux * pi * x / L) * a_ux * pi / L;
grad_u_an[1] = -u_y * sin(a_uy * pi * y / L) * a_uy * pi / L;
grad_u_an[2] = -u_z * sin(a_uz * pi * z / L) * a_uz * pi / L;
grad_v_an[0] = -v_x * sin(a_vx * pi * x / L) * a_vx * pi / L;
grad_v_an[1] = v_y * cos(a_vy * pi * y / L) * a_vy * pi / L;
grad_v_an[2] = v_z * cos(a_vz * pi * z / L) * a_vz * pi / L;
grad_w_an[0] = w_x * cos(a_wx * pi * x / L) * a_wx * pi / L;
grad_w_an[1] = w_y * cos(a_wy * pi * y / L) * a_wy * pi / L;
grad_w_an[2] = -w_z * sin(a_wz * pi * z / L) * a_wz * pi / L;
```

References

Roy, C., T. Smith, and C. Ober (2002). Verification of a compressible cfd code using the method of manufactured solutions. In *AIAA FLuid Dynamics Conference and Exhibit*, Number AIAA 2002-3110.